1. Identify the Attacked Endpoint

Create a Bash script to find the endpoint (URL) that received the most requests, indicating it was likely the target of the attack.

Functionality:

- Extract the requested URLs from the log file.
- Count the occurrences of each endpoint and identify the most frequently requested one.

```
TIP: try to find where the most request have been sent
```

```
┌──(oumaimaⓀhbtn-lab)-
[…/web_application_security/0x0b_web_application_fast_incident_response]
└─$  ./1-endpoint.sh logs.txt
/
```

```
#!/bin/bash
awk '{print $7}' $1 | sort | uniq -c | sort -nr | head -n 1 | awk '{print
$2}'
```

Breakdown of the Commands:

1. **`awk '{print $7}' $1`**:
   - This uses `awk` to extract the **7th field** from each line of the log file. In Apache or Nginx log formats, the 7th field typically corresponds to the requested URL or endpoint (the part after the domain name in the request). For example:

     ```
     GET /about HTTP/1.1
     ```

     In this case, `/about` is the requested URL, which we want to count.
2. **`sort`**:
   - This command sorts the extracted URLs alphabetically. Sorting is required before we can count the occurrences of each unique endpoint.
3. **`uniq -c`**:
   - The `uniq -c` command counts how many times each URL appears in the sorted list.
     - For example, if `/about` appears 5 times, the output will be `5 /about`.
4. **`sort -nr`**:

- This command sorts the counts in **numerical reverse order** (from highest to lowest). This way, the most requested endpoint appears at the top.

5. `head -n 1`:
   - This command returns only the first line, which contains the most requested endpoint.
6. `awk '{print $2}'`:
   - Finally, this extracts the second field from the output of the previous command, which is the actual endpoint URL (ignoring the count).

Example Log File (`logs.txt`):

```
192.168.0.1 - - [10/Nov/2024:10:00:00 +0000] "GET /about HTTP/1.1" 200
192.168.0.2 - - [10/Nov/2024:10:00:05 +0000] "GET /contact HTTP/1.1" 200
192.168.0.1 - - [10/Nov/2024:10:01:00 +0000] "GET /about HTTP/1.1" 200
192.168.0.3 - - [10/Nov/2024:10:01:10 +0000] "GET / HTTP/1.1" 200
192.168.0.1 - - [10/Nov/2024:10:01:30 +0000] "GET /about HTTP/1.1" 200
```

How the Script Works:

1. `awk '{print $7}' logs.txt` extracts the requested URLs:

   ```
   /about
   /contact
   /about
   /
   /about
   ```

2. `sort` sorts them alphabetically:

   ```
   /about
   /about
   /about
   /contact
   /
   ```

3. `uniq -c` counts the occurrences:

   ```
   3 /about
   1 /contact
   1 /
   ```

4. `sort -nr` sorts by count in descending order:

   ```
   3 /about
   1 /contact
   1 /
   ```

5. `head -n 1` gets the most requested endpoint:

```
3 /about
```

6. `awk '{print $2}'` extracts just the endpoint:

```
/about
```

Final Output:

```
/about
```

This output indicates that the `/about` endpoint was the most requested, likely the target of a DoS attack.

Example Usage:

```
$ ./1-endpoint.sh logs.txt
/about
```